Provincial Events – Developers Guide

Kevin Turner   Jan 2016

Contents

Pre-requisites

Working on the assumption that we are working with a blank canvas, the following things must be installed on any system (regardless of platform):

MySql

This comes pre-installed on many hosted linux servers (like Plesk Centos).

If you don't have it installed then you need to do a platform specific install of MySql by following the relevant downloads and instructions from http://dev.mysql.com/downloads/

If you are a developer working on a Windows PC it is very useful to obtain MySql by simply downloading and installing XAMPP from https://www.apachefriends.org/index.html

With this you get lots of goodies, including Apache, PHP and PHPMyAdmin for administering your MySql databases. We don't actually need Apache for Square Events but it is useful to have.

Node

Install the latest version of Node from https://nodejs.org/en/download/

Git

Used for all source change management: https://git-scm.com/downloads

Background reading: http://readwrite.com/2013/09/30/understanding-github-a-journey-for-beginners-part-1

Redis

Redis is an in-memory data store that can optionally used as a more efficient way to manage sessions. The application can be configured to use redis (and does so by default) or just use RAM for sessions. Redis should be downloaded and installed from https://redis.io/ for most platforms but should be obtained from https://github.com/MSOpenTech/redis/releases

GitHub

If you have not already got an account, register at https://github.com/

An important URL for future use is the repo(sitory) for the Square Events application: https://github.com/kpturner/sails_events.git

Windows

If you are a developer working on a Windows PC, the following tools are useful additions to your armoury for developing and testing:

TortoiseGit

By no means an essential tool but quite useful for a Windows Explorer GUI interface to Git (which is primarily a command line tool): https://tortoisegit.org/

PuTTY for Windows

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Visual Studio Code

If you don’t already have a favourite editor, this is a very nice lightweight editor to use with many languages: https://code.visualstudio.com/

If you install this, be sure to allow it to add an option to windows explorer to enable you to open resources with vs code from the context menu.

We also need to make a small config change:

Edit the following file

```
C:\Program Files (x86)\Microsoft VS Code\resources\app\extensions\html\package.json
```

Change the “extensions” JSON property so it looks like this:

```
"extensions": [ ".html", ".htm", ".shtml", ".mdoc", ".jsp", ".asp", ".aspx", ".jshtm", ".rml", ".ejs" ],
```

Copy

```
C:\Program Files (x86)\Microsoft VS Code\resources\app\extensions\html
```

to

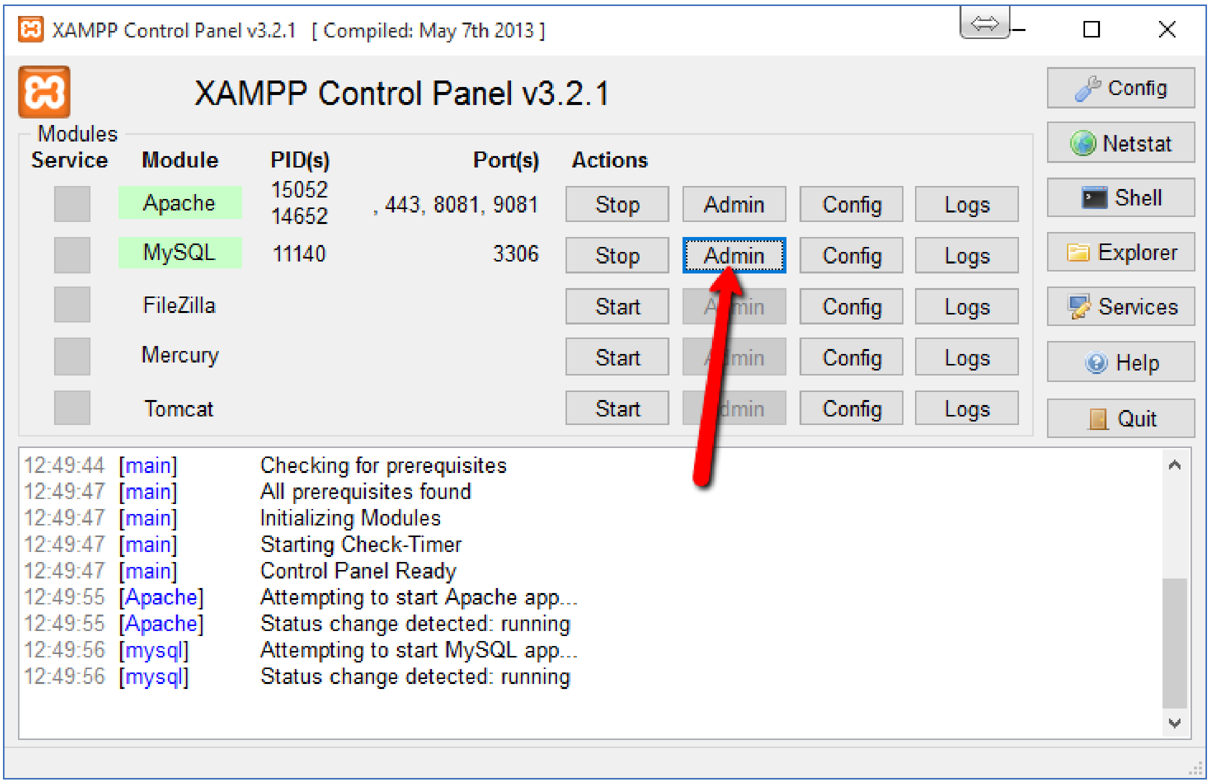
```
C:\Users\<user>\.vscode\extensions
```

Setting up a development environment

Database

First of all you need to create an empty MySql database called **squareevents** and a user with all necessary privileges to that database called **squareevents@localhost** with a password of <obfuscated>.

If you have downloaded XAMPP then then easiest and quickest way is to use PHPMyAdmin to set this up. From the control panel, you can start MySql and then click "Admin" to get into PHPMyAdmin:


XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]
XAMPP Control Panel v3.2.1
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
	Apache	15052 14652	, 443, 8081, 9081	Stop	Admin	Config	Logs
	MySQL	11140	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
12:49:44  [main]     Checking for prerequisites
12:49:47  [main]     All prerequisites found
12:49:47  [main]     Initializing Modules
12:49:47  [main]     Starting Check-Timer
12:49:47  [main]     Control Panel Ready
12:49:55  [Apache]   Attempting to start Apache app...
12:49:55  [Apache]   Status change detected: running
12:49:56  [mysql]    Attempting to start MySQL app...
12:49:56  [mysql]    Status change detected: running
```

Get the source code

Start up a command prompt window and as a one off exercise, register yourself with your preferred name and email address:

```
git config --global user.name "Your Name Here"
git config --global user.email "your_email@youremail.com"
```

After this is complete, navigate to your default document location which, in windows, will be *c:\users\<user>*

From here create the following directory structure and navigate there:
c:\users\<user>\sails\projects

Whilst in the *\sails\projects* directory, we will clone the git repo from GitHub like so:

```
git clone https://github.com/kpturner/sails_events.git events
```

Navigate to the location of the cloned repo in a shell/command prompt

The next thing to do is make sure all the node packages required are installed on the system. Assuming you have installed node successfully, we will start with globally installing http://sailsjs.org/ using NPM

Key in the following (very important to specify the –g option):

```
npm install sails -g
```

Once this is installed, we will install all the packages locally required by the square events application:

```
npm install
```

We will also need forever.js to run the app continuously (it will auto-restart after failure or upgrade).

```
npm install forever -g
```

Starting the application

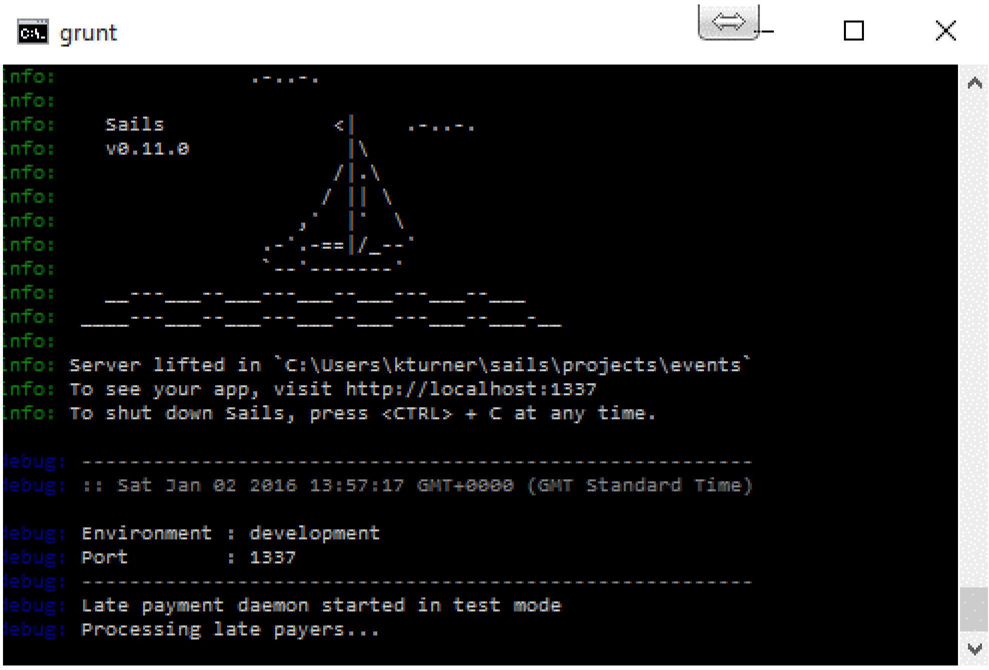
Assuming you are still in the *c:\users\<user>\sails\projects\events* directory, we can start the application. This will also automatically build\update the database, assuming you have already created the empty **squareevents** database and created the **squareevents@localhost** user with a password of <obfuscated>.

From a command line

Use the following command to fire everything up:

```
sails lift
```

All being well:



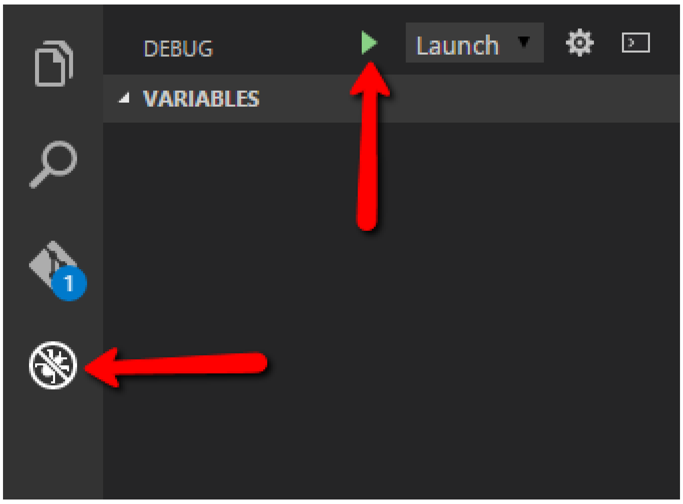
```
info:                .-..-.
info:
info:    Sails              <|    .-..-.
info:    v0.11.0             |\
info:                       /|.\
info:                      / || \
info:                    ,'  |'  \
info:                 .-'.-==|/_--'
info:                 `--'-------'
info:    __---___--___---___--___---___--___
info:  ____---___--___---___--___---___--___-__
info:
info: Server lifted in `C:\Users\kturner\sails\projects\events`
info: To see your app, visit http://localhost:1337
info: To shut down Sails, press <CTRL> + C at any time.

debug: --------------------------------------------------------
debug: :: Sat Jan 02 2016 13:57:17 GMT+0000 (GMT Standard Time)

debug: Environment : development
debug: Port        : 1337
debug: --------------------------------------------------------
debug: Late payment daemon started in test mode
debug: Processing late payers...
```

From within VS Code (for debugging)

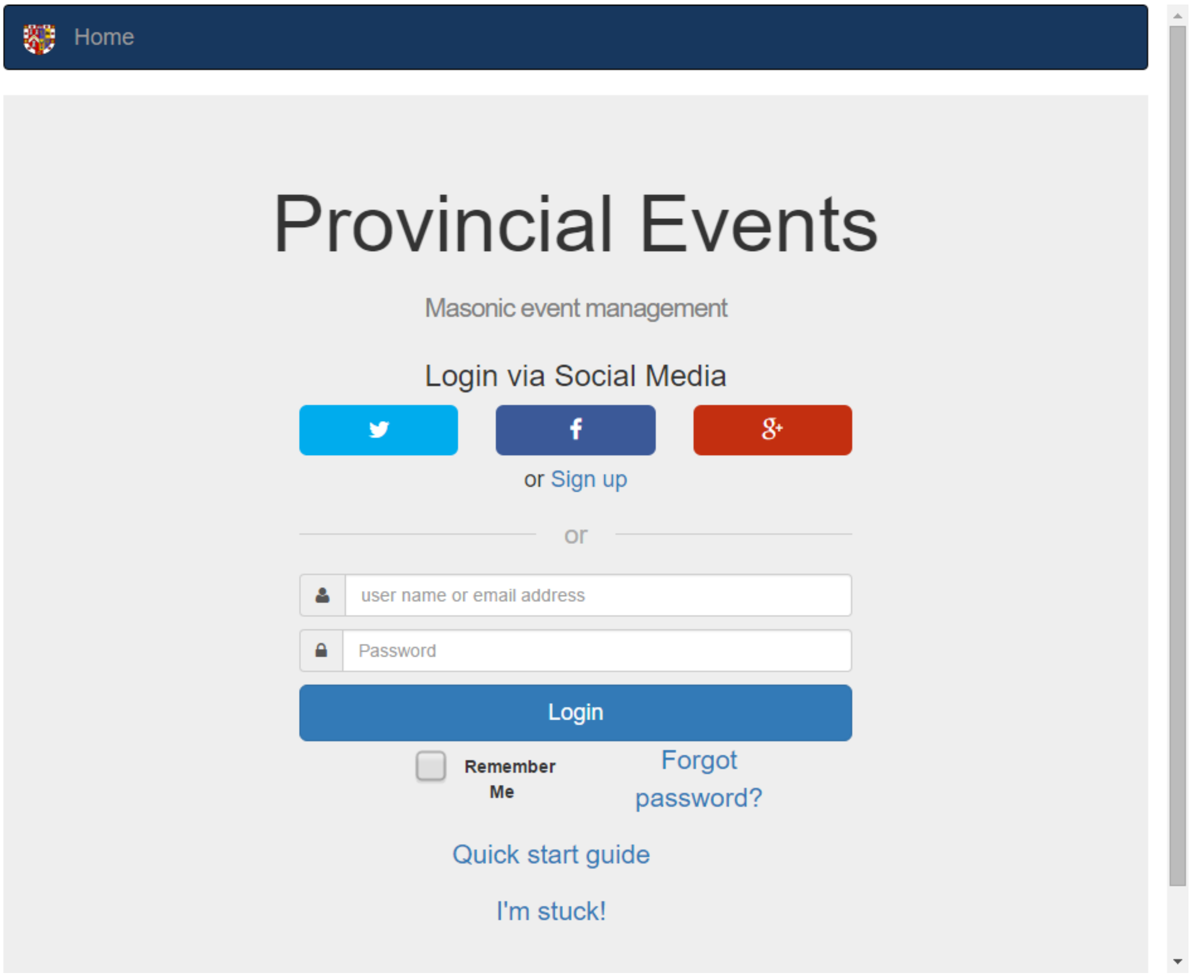
Click on the “debug” icon in the left hand menu and then click on the Start button at the top using the Launch script:



Further information: https://code.visualstudio.com/docs/editor/debugging

Running the application

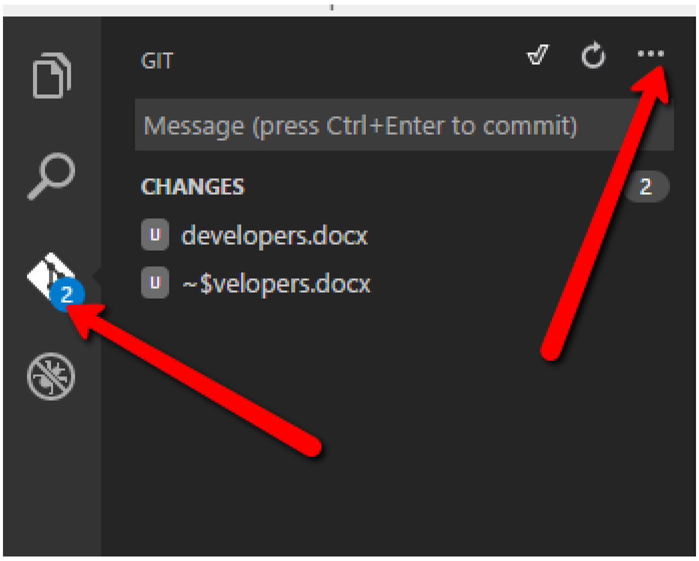
You should now be able to run http://localhost:1337 and see the application fire up in your default browser:



Editing the code

Assuming you will use VS Code to edit the code, the easiest thing to do is to use windows explorer to navigate to C:\Users\<user>\sails\projects folder, right click on the events folder and take the “Open with code” option.

To pull updates from GitHub you can click on the Git interface button in the left hand options then, from the menu in the top right hand corner, and take the “pull” option. This will pull the latest updates from GitHub.



GIT
Message (press Ctrl+Enter to commit)
CHANGES
2
U developers.docx
U ~$velopers.docx
2

Putting changes into production

Currently the application is running on my own Centos Server (running Plesk) hosted by Poundhosts.

The hostname is kpturner.co.uk

I use PuTTY to deploy changes and start/stop the server. In windows you can just run putty.exe and then open an SSH connection to kpturner.co.uk on port 22.

Login with the supplied user id and password (to be provided) then:

Change to correct location:

```
cd /usr/sails/projects/events
```

Pull latest updates from GitHub:

```
bash gitupdate.sh
```

Stop the service:

```
service events-service stop
```

The next two commands are only required if database updates need to be implemented:

```
sails lift
ctrl-c
```

Start the service again:

```
service events-service start
```

Notes on setting up a Linux server

Look at the **installing_on_centos.txt** file located in the *sails*\projects\events folder.

Notes on creating the readme.md markdown file for github

If you change the documentation (readme.docx) you must also save it as a PDF. Additionally, use must create a markdown file for GitHub called readme.md.

To do this, make sure you have a tool called pandoc installed: http://pandoc.org/

Then navigate to the “events” project folder that contains readme.docx and run this command:

```
pandoc -s readme.docx --wrap=none  --extract-media=readme_media -o readme.md
```

At the time of writing this creates image references like this:

```
![](readme_media/media/image1.png){width="3.9057622484689416in" height="6.947048337707787in"}
```

And this is causing problems so I manually removed the bits highlighted in yellow. I haven’t had time to work out how to stop this.

Notes for authorising URLs for passport login

Google: https://console.developers.google.com Look at credentials section for the app

Facebook: https://developers.facebook.com/ Go to the app settings, “Advanced” tab and find “Valid OAuth redirect URIs”

Twitter doesn’t seem to check the redirect URI